NASA-TM-111722

AUTOMATION OF SOME OPERATIONS OF A WIND TUNNEL USING ARTIFICIAL NEURAL NETWORKS

Arthur J. Decker* and Alvin E. Buggele**

NASA Lewis Research Center

ABSTRACT

Artificial neural networks were used successfully to sequence operations in a small, recently modernized, supersonic wind tunnel at NASA-Lewis Research Center. The neural nets generated correct estimates of shadowgraph patterns, pressure sensor readings and mach numbers for conditions occurring shortly after startup and extending to fully developed flow. Artificial neural networks were trained and tested for estimating: sensor readings from shadowgraph patterns, shadowgraph patterns from shadowgraph patterns and sensor readings from sensor readings. The 3.81 by 10 in. (0.0968 by 0.254 m) tunnel was operated with its mach 2.0 nozzle, and shadowgraph was recorded near the nozzle exit. These results support the thesis that artificial neural networks can be combined with current workstation technology to automate wind tunnel operations.

*Senior Research Scientist, Instrumentation and Control Technology Division

**Operations Engineer, Aeropropulsion Facilities and Experiments Division

Member AIAA

21000 Brookpark Rd.
Cleveland, Ohio 44135

INTRODUCTION

Automation of wind tunnel operations using artificial neural networks has been tested in a 3.81 by 10 in. (0.0968 by 0.254 m) subsonic/transonic/supersonic wind tunnel at NASA-Lewis Research Center. A possible approach to automation, which uses operator-trained, workstation-resident, software neural nets as sequencers, was reported previously by the authors.[1,2] One of the reports discussed the use of archival flow visualization records from the Lewis tunnel to train the neural networks.[1] This paper will discuss the performance of neural net sequencers which were trained and tested on recent runs of the wind tunnel with its mach 2 nozzle and will indicate why and where said sequencers should be useful. Briefly stated, a sequencer estimates for a wind tunnel the next appropriate control settings, flow visualization patterns, or sensor values from the current control settings, flow visualization patterns, or sensor values. A distributed or programmable logic control system[3] can then use these estimates to set the next appropriate state of the tunnel without human intervention.

The authors reported previously that neural net sequencers performed poorly for some subsonic/transonic/supersonic cascade configurations.[2] Recent results using the small tunnel's mach 2 nozzle, by contrast, have been fair to very good in selected flow regions. This improvement is attributed to: a more repeatable, cleaner flow visualization setup, better tunnel sidewall stability and the increased experience of the tunnel operator in preparing training sets for neural net sequencers. Hence, the results reported herein refer to the mach 2 configuration only.

There are several motives for considering the automation of wind tunnel operations using neural nets in particular. First, neural net sequencers can replace personnel whose job is to execute routine sequences of actions during tunnel start-up and operation. A major reason for using human operators for some routine tasks is that those tasks require the interpretation of patterns such as flow visualization. Neural nets perform these interpretations quite effectively as will be demonstrated. Second, neural nets can assure the quality of operations. The nets can be trained with the actual examples of the most expert personnel, as reported in this paper, or with model generated data. Experimental protocols and wind tunnel characteristics can be interfaced efficiently. Third, neural nets are a tool for managing and monitoring the operation of a wind tunnel. Neural nets measure the utilization, effectiveness, and consistency of performance of all inputs including: personnel, instrumentation, flow conditioning, operations methods and services. An input is easily removed from a training set, or added to a training set, for evaluation of its necessity or effectiveness. Fourth, neural nets can provide continuous monitoring of the health of the wind tunnel. The neural nets will estimate both the current and the next flow visualization or sensor values. Significant deviations from the actual flow visualization and sensor values can be detected. Finally, neural nets are an effective way to use increasingly powerful workstations. The neural nets--resident in the SGI Crimson workstation used for the work discussed herein--process more than 300 flow visualization records per second.[2] Hence, 10 to 100 nets could be used to process different groups of sensors, flow visualization pictures, and valve settings, even if the tunnel state were changed as often as once per second. Software is also under development to allow the workstation to communicate control settings to the tunnel's distributed or programmable logic control system.[3]

Automation of wind tunnel operations is targeted toward three classes of tunnels. Development and experimentation are being done in a wind tunnel described in this paper. That tunnel has many of the features of the larger tunnels at Lewis Research Center, but its small size makes it an affordable, practical place for testing automation technology. A second targeted class is the collection of existing large wind tunnels. A practical problem here is that neural nets are trained by example. A tunnel must be run to generate the examples, and the cost effectiveness of that approach is questionable. Hence, the effective application of automation to this target probably will require identification of one or more of the following situations: operations which use pattern identification or visualization and occur commonly for many different kinds of tests; other kinds of operations that are executed frequently and repeatedly; and situations where models of tunnel operations exist and can be used to supplement limited data. The massive parallel processing capabilities of neural nets can also be supplemented and augmented to some extent with the rule insertion capabilities of expert systems, fuzzy control, or simple programming logic. Finally, new wind tunnels are targets for automation, if their designs can be constrained to use fully the automation capabilities of artificial neural networks and other artificial intelligence technologies as well as the inputs (particularly flow visualization) that those technologies require.

The results in this paper provide evidence to support two of the motivations for using artificial neural networks for automating wind tunnel operations including: the efficient, accurate interpretation of pattern data as exemplified by flow visualization patterns; and the concept of training a neural network with an expert's examples. First the wind tunnel, the shadowgraph flow visualization system, the workstation, the neural-net sequencers, the data handling system and the control system are described. Next, the experimental procedures are explained where image-to-sensor, image-to-image, and sensor-to-sensor neural-net sequencers are trained and tested. Then, the results are discussed, and the ranges for good and poor performance are presented and explained.

MODERNIZED WIND TUNNEL--FOR AUTOMATION

The wind tunnel used to generate the training sets for the neural-net sequencers was actually an upgrade of the first supersonic wind tunnel operated at Lewis Research Center (circa 1946).[4] The original tunnel, Fig. 1, was essentially an 8 ft (2.44 m) long, 4 by 10 in. (0.102 by 0.254 m) rectangular duct connected to a central air (C.A.) supply via a 12 in. (0.305 m) diameter valve and to an altitude exhaust system via a diffuser and a 42 in. (1.07 m) diameter exhaust valve. The tunnel has been upgraded many times since 1946. Recently, the tunnel was modernized to have many of the same systems as are used to control the operations of large wind tunnels. The tunnel is now suitable as a multi-user platform for testing instrumentation and components as well as developing parallel processing and control strategies. A neural-net experiment, performed with a cascade configuration of the tunnel, has been reported.[2] The modern mach 2 configuration used to generate the training sets for this paper is shown in Fig. 2. This configuration was upgraded considerably in the last two years, as shown in Fig. 3, to provide heated, filtered, inlet air with variable flow conditioning, boundary layer bleeds, and dynamic shock positioning and shape control up to 100 Hz. The modern tunnel's inlet supply also has several additional 8 in. (0.203 m) diameter valves, and the tunnel's exhaust system has an additional 8 in. (0.203 m) diameter bypass valve.

The specific components of the tunnel needed to interpret the experimental procedure and results discussed in this paper are shown in the highly simplified diagram of Fig. 4. Flow is essentially controlled through the combustion air valves AC-2408 A and AC-2408 B. The groups of boundary layer bleed and exhaust valves were preset for training set generation. In fact, any serious attempt at complete automation would need to recognize that the tunnel, as shown in Fig. 3, uses more than 60 valves.

Flow visualization was acquired through a pair of 47 in. (1.19 m) long windows. The shadowgraph field for this work was actually centered near the right edges of the windows 40 in (1.02 m) downstream of the nozzle throat or approximately 2 in. (0.05 m) downstream of the nozzle exit. The positional stability and repeatability of the one-pass shadowgraph and the tunnel wall were assured to within 0.002 in. (0.005 cm) using a hard mounted system with taper dowel pins. The shadowgraph carriage can be relocated within this accuracy along the entire length of the tunnel. The neural nets require this relocation accuracy as has been reported.[2] The actual shadowgraph field recorded was about 10 by 8.5 in. (0.25 by 0.21 m). A sample image is inserted in Fig. 4 for reference.

Figure 4 shows only three of many sensors. The static inlet or plenum pressure is indicated by P_7, and a static test section pressure located 0.5 in. (1.27 cm) downstream of the nozzle exit is indicated by P_8. The mach number probe MN_3 was located 48 in. (1.22 m) aft of the nozzle throat or about 8 in. (0.2 m) downstream of the nozzle exit, just downstream of the developing test rhombus. The pressure sensors are, in fact, part of a 186 channel electronically sensed pressure system with an accuracy of ±0.033 psi (±228 pascal). Considerably greater pressure fluctuations than the accuracy are induced at times by the upstream or downstream hardware; the time averages of these fluctuations rather than the accuracy of the transducers determine the tunnel operator's responses and neural-net training sets.

Tunnel information is handled, and control is exercised, with the systems to be described in the next section.

TUNNEL INFORMATION HANDLING AND CONTROL

The subsystems used to acquire, handle and process information on the state of the tunnel as well as exercise control of the wind tunnel are outlined in Fig. 5. The SGI Crimson XS24 workstation is a key element and is representative of the growing computer technology that makes this study relevant. The workstation has 256 megabytes of random access memory for compilation and high speed execution of the neural net algorithms as well as four VME slots, two SCSI buses, an Ethernet connection, and four serial ports for communications. The VME slots, Ethernet connection, and serial ports are used to acquire and change the state of the tunnel. All flow visualization patterns enter the computer via a VME mounted, color video frame grabber.[5] The frame grabber supports a variety of television and image formats, but the shadowgraph patterns discussed herein were acquired with a NTSC, 646 X 486 pixel, black-and-white CCD camera.[6] This camera operates at 30 frames per second with two fields per frame; hence unsteady flow visualization patterns are time averaged and may change from one field to the next. At this stage, all sensor and control data (pressures, temperatures, valve settings, and mach numbers) are acquired by Lewis's ESCORT D central data acquisition system. The data is transmitted in a few seconds to the workstation's Ethernet connector, via a local area network, in the form of possibly more than 800 ascii encoded sensor and control values. Fast acquisition of sensor values and transmission of control information require direct communications with the facility's MODICON programmable logic control system.[3] A software package is being developed to implement direct communications with MODICON through the workstation's serial ports. A noteworthy point is that the workstation is intended eventually to be used as a slave module in the control system. Image handling, massive parallel processing, and neural-net decision making will be done in the workstation, but the results will be filed for use at the convenience of the control system. This approach makes it easier to interface automation to existing protocols and distributed control systems in wind tunnels and other facilities.

A commercial package was used to create the workstation's artificial neural networks.[7] This package contains versions of a large number of neural net paradigms, but the work discussed herein used only the feedforward neural networks (exemplified by Fig. 6), Fuzzy ARTMAP, and the modular neural network.

At this point, the authors wish to emphasize that successful automation of wind tunnel operations depends strongly on a practical, personal, competent knowledge of the wind tunnel, its instrumentation, and its controls and only very weakly on a knowledge of neural network technology. An extensive discussion of the theory and practice of neural networks is beyond the scope of this paper and is not especially necessary or helpful. There is an extensive literature discussing artificial neural networks including books[8,9] as well as tutorials and workshops.[10,11]

Speaking very briefly, an artificial neural network as depicted in Fig. 6 is a collection of interconnected nonlinear processing elements or nodes. (Figure 6 shows full connections for only one input node and one output node.) The connections between nodes typically are weighted; the network typically has vectors or arrays of inputs and outputs. The program or mapping between inputs and outputs typically is encoded in the connection weights. The algorithm for adjusting the weights to achieve a desired mapping of inputs onto outputs is called a training algorithm. Biological allusions such as "neural net" and "training" originated in the weak similarities of some artificial neural networks to the living variety. For those allusions, connection weights are associated with the strengths of electrochemical synapses; nodes are associated with neurons; and the output connections of nodes are associated with axons. Such semantics have no direct application in this paper.

As explained, the objective of this work was to train or program a neural network or system of neural networks to estimate the next desired state of the wind tunnel from its current state. The inputs to the neural networks then consisted of vectors of pixel values from the CCD camera, pressure sensor readings, mach numbers, and valve settings. The outputs consisted of exactly the same kinds of information, but represented the next operating state of the wind tunnel. The expert operator generated a representative training set of input-vector, output-vector pairs during a run; that is, the operator developed a training set by example. Said training pairs must cover the space of likely tunnel states or transitions between states for successful training.

We considered among the following factors in selecting neural net software packages and paradigms for learning the training sets. The first considerations were convenience and cost: we used commercially available, menu driven packages to avoid incurring software writing costs. The commercial package used for the work reported herein[7] provided menu driven control of all processes including: training set selection, test set selection, net selection, net creation, net training, net testing, monitoring of net performance, and C language encoding of the trained net. The package allowed the creation of nets containing up to 20,000 nodes or neurons. A second consideration was the ability of a net to generalize or to extrapolate and interpolate given the discrete examples of the training set. Possible additional considerations were the training time and the recall time or response time of the trained net.

The feedforward net depicted in Fig. 6 is a common example of neural networks; feedforward nets were used to generate most of the results in this paper. The nodes for the net of Fig. 6 are arranged in three layers. The lowest layer or input layer in this example contains 3,648 inputs to receive the pixel values from 57 X 64 pixel shadowgraph images. (The 8 bit pixel values must actually be normalized between 0 and 1.) The second layer, containing 16 nodes, is called a hidden layer. Each node in the hidden layer is connected to every node in the input layer via weighted connections . Each node in the hidden layer also has a bias input not shown. The weighted inputs are generally summed by the hidden layer node, and the sum is transformed using a nonlinear function. The transformed sum is then applied to the output of the node, and the output is fanned out to the highest layer or output layer. Each node in the output layer performs similar processing. The output layer also contains 3,648 nodes corresponding to a 57 X 64 pixel output image. The output image might be the next image in a sequence of images associated with a sequence of tunnel states. Training is an iterative process where input images are applied to the inputs; output images are generated and compared with the training image; and the weights are adjusted. There are many kinds and variations of training algorithms; the reader is referred to the references for discussions.

Nets such as that shown in Fig. 6 currently require a workstation environment. The menu driven version of the trained net occupies 5 to 6 megabytes of storage. The C language code with comments for the trained net also occupies 5 to 6 megabytes. The compiled code and calling program occupy 3 to 4 megabytes. But, compilation requires more than half of the 256 megabyte random access memory. Execution speed is more than adequate. One compiled net was measured to process more than 300 images per second where the images were 60 X 60 pixels and where there were 7 output nodes for 7 sensor values.

The tunnel operator currently operates the tunnel with manual controls while monitoring the sensor values and flow visualization. There are more than

60 valves as has been mentioned, and most are not operated during a run. Many are preset during the tunnel's initial tuning phase to provide proper boundary layer bleed conditions. The major concern during operation of the tunnel for any purpose, including training set generation, is the integrity of the two one-of-a-kind optical windows shown as the longer windows in Fig. 2. Pressure changes are not permitted to exceed 3 to 6 psi (20 to 40 kilopascals) per minute. The major consequence of this requirement is that operations are performed slowly, and even the local area network is fast enough for the neural-net experiments. The actual experimental procedure is described next.

EXPERIMENTAL PROCEDURE

The experimental procedure includes training set acquisition, training set preprocessing, neural net training, and neural net testing. Training set acquisition consists of recording the sequence of tunnel states leading to a particular state such as fully developed flow. The objective is to select exactly those sensor readings, control settings and flow visualization patterns that the operator refers to in setting the tunnel state. Some combinations of those sensor readings, control settings, and visualization patterns are then used to construct the input and output vectors for the neural networks.

The actual procedure for training set acquisition was more poorly defined. The operator was forced to learn by trial and error a procedure for mach 2 operation of the modernized tunnel, and sequences varied significantly from one run to the next. The stability of the flow varied greatly from start-up to fully developed flow. Pressure fluctuations more than 1 psi (7 kilopascal) were encountered at times. Some flow regions produced very stable shadowgraph patterns; other flow regions produced time averaged patterns. The ESCORT D data dumps displayed all the sensor values, but contained only snapshots of possibly fluctuating data. The central air handling services for the tunnel were shared by a large number of facilities; the tunnel did not have a good system for regulation. Hence, the sensor values for a given flow visualization pattern often varied from day to day, hour to hour, or even minute to minute.

This section will discuss the procedures for training set development from two runs. For one run, the operator viewed certain pressure sensor readings, valve settings, and mach numbers, and set the next values of the same, to generate the training sequences. For the second run, the operator viewed only the shadowgraph patterns to generate shadowgraph-to-shadowgraph sequences. The two kinds of runs in fact were performed in different months.

For this paper, neural networks were trained with the sensor readings and shadowgraph patterns from the second (visualization controlled) run; the accuracy of training was measured; and the trained nets were tested against the first (sensor controlled) run. The procedures for performing the sensor and visualization controlled runs of the tunnel and acquiring the training data are discussed next.

The following procedure was adopted to protect the optical windows and acquire the training data. The test section was evacuated through the boundary layer bleeds from ambient pressure to a pressure of 2.0 psia (13.8 kilopascal). The sidewall, roof, floor, and pre-throat bleeds were opened in the stated order to control the evacuation rate to 3 to 6 psi (20 to 40 kilopascal) per minute to protect the windows. Several minutes were then needed for automatic calibration of the data acquisition system and the recording of baseline shadowgraph. For the sensor controlled run, the inlet or plenum pressure P_7 was increased in approximately 1 psi (6.9 kilopascal) increments. Shadowgraph and a data dump were recorded after each increment. Approximately 20 minutes were required to reach a final plenum pressure of about 35 psia (241 kilopascal). A similar, but reversed, schedule was followed for shutdown. The operator was able to perform the visualization controlled run by remembering and distinguishing changes in the sequence of shadowgraph patterns corresponding to approximately 1 psi (6.9 kilopascal) increments. The valves were operated to change from one pattern to the next. Shadowgraph and a data dump were recorded for each pattern in the sequence.

As stated, all training reported in this paper was accomplished with the visualization controlled run even though the operator did not use the sensor data to control that run. Part of a sequence of shadowgraph images from the visualization controlled run is shown in Fig. 7. The top row in that figure shows 646 X 486 pixel images. The flow is from right to left. The shadowgraph fields occupy only part of the images. The images were prepared for training first by cropping them to 340 X 302 pixels to show only the shadowgraph fields. The resolution was then reduced to 64 X 57 pixels for handling by the neural net software, and the images were converted to binary format. These operations were accomplished using standard SGI image processing commands. An image-to-image training set was then constructed simply by combining adjacent shadowgraph patterns in the sequence as input-output pairs. The actual format depends somewhat on the protocols of the commercial neural net package and on the neural net type. A feedforward net, for example, requires that the input pixels, whose values can be between 0 and 255, be normalized between 0 and 1 . For this paper, all pixels were normalized for the full range of 0 to 255 . An alternative is to note the actual range of values exhibited by each individual pixel in the full training set. The pixels are then individually normalized for their particular ranges. Individual pixel normalization was very sensitive to time varying illumination effects as well as time varying window patterns and was not effective. The output pixel values were normalized between 0.2 and 0.8 to accommodate the node nonlinear transfer function (a sigmoidal function).

Training set preparation using the pressure sensors, mach numbers and valve settings was more difficult. Shadowgraph patterns were definitive; but as mentioned earlier, pressures, downstream mach numbers and supply-air valve settings depended on the other users and the configuration of the air handling system. The valves have nonlinear hysteresis effects. Hence, a mapping between a particular shadowgraph pattern and corresponding pressure settings, mach numbers and valve settings is not definite. It is necessary to look for invariants. The results reported herein are based as in Fig. 4 on: the inlet pressure P_7, one test section pressure P_8, and the downstream mach number MN_3 . The input or output vector actually has the form: $[P_7, \ (MN_3)^2 P_8]$. The performances of the nets using P_8 or MN_3 separately were very poor for the reasons mentioned. The performances, using the product $(MN_3)^2 P_8$ forming the second component of the vector, were fair to good as will be seen. The sensor values listed here are a small fraction of the total number available. But P_7, P_8, and MN_3 were interrogated often by the tunnel operator, and were therefore relevant to the objective to train by example.

Adjacent vectors in the operations sequence were then combined to form the sensor-to-sensor training sets. Again, the training sets reported in this paper were derived from the image (shadowgraph) controlled run. The operator could not see or examine the sensor values while performing that run; the sensor values were recovered from the data dumps later. The pressure range for normalizing P_7 is 0 to 50 psi (0 to 345 kilopascal), and the range for normalizing $(MN_3)^2 P_8$ is 0 to 24 psi (0 to 165 kilopascal).

An image-to-sensor sequencer is also discussed herein. For training, the 64 X 57 pixel images were used as inputs. The outputs consisted of the current sensor pair and the next sensor pair in the sequence, where a sensor pair is the vector defined two paragraphs above.

The neural net for the image-to-image sequencer was essentially that shown in Fig. 6 with 3648 input nodes, 16 hidden nodes, and 3648 output nodes. A modified back propagation algorithm was used for training. Good training requires about 3000 iterations per training record. A training record is one of the input-output image pairs.

The neural net for the sensor-to-sensor sequencer was a feedforward net with 2 input nodes, 6 hidden nodes, and 2 output nodes. The image-to-sensor net had 3648 input nodes, 16 hidden nodes, and 4 output nodes.

The neural nets were tested first against their training sets and then against test sets from the sensor controlled run acquired a month earlier. Test sets are formatted exactly the same as training sets.

The runs were divided somewhat arbitrarily into three regions for this exercise. Start-up involved unstable normal shock-train patterns. The transition region showed stable shock patterns whose shapes and positions varied significantly as the flow was changed to a fully developed flow in the test rhombus. The fully developed region contained stable shock patterns that changed only slowly as the inlet pressure was increased significantly. Nets were trained with training sets developed from all three regions, but most training records originated from the transition region.

RESULTS AND DISCUSSION

The neural nets outlined in the previous section were trained with records from the image (shadowgraph) controlled run and tested with records from the sensor controlled run. The results from that training and testing are presented in this section.

In fact, a large number of nets were trained and tested on various runs during a period of about a year. Most of these runs indicated the need to solve problems such as assuring the accuracy of alignment of the flow visualization system[2] or responding to the effects of random changes in the central air handling system. No two run sequences were identical, because of varying demands on the central air handling system, which makes it difficult to compare runs. However, the operator had developed a consistent, personal protocol for bringing up the tunnel for the two runs to be compared. There was no constraining model which forced the operator to perform identically during the two runs. Identical performances would be fortuitous; since the operator looked at pressure sensors for the sensor controlled run and shadowgraph for the image controlled run. The two runs are compared by looking for state-to-state transitions that overlap in the two runs as will be explained.

Tables I and II list the output records for training and testing, respectively, the image-to-sensor net; the inputs are the corresponding 64 X 57 pixel shadowgraph images. Table I lists the sensor values for an 11 record training set from the image (shadowgraph) controlled run. Each output record contains an arbitrary state number, the current sensor values, and the next sensor values for that state. The sensor values again are the inlet or plenum pressure P_7 and the grouping $(MN_3)^2 P_8$ which contains the downstream mach number MN_3 and the test section pressure P_8 . The ratio r of these quantities is also listed for quick reference. That ratio is an approximate invariant, and can be used to identify where the current-to-next tunnel-state transitions overlap. This feature is important; since the actual sensor values for a given shadowgraph pattern depended on the configuration of the central air handling system and its users.

Table II lists the sensor values for a 12 record test set from the earlier sensor controlled run. The format of Table II is the same as the format of Table I.

Figure 8 shows the performance of the image-to-sensor net in predicting the current sensor values. The top row of that figure shows the response of the net to the training set, and the bottom row shows the response of the net to the test set. The solid lines in the plots are the data-dump sensor values, and the solid dots are the net generated sensor values. The net is seen to generate the training values well (upper row of Fig. 8), but is seen to generate the test values well only in the middle regions of the lower-row plots.

Figure 9 has the same form as Fig. 8 , but shows the next sensor values in sequence rather than the current sensor values for a state. This plot is more appropriate for predicting the control potential of the net.

There are no identical transitions in Tables I and II; however there is overlap in both the transitions between ratios and the trend in those transitions beginning at about state 4 in Table II. The net generates the test values well beginning at about this state. Lack of overlap is not the only factor in explaining the poorer performance at the lower pressures (smaller state numbers). Flow fluctuations are large at the lower pressures due to disturbances caused by partly open control valves. The shockwave jitter is as much as ± 1/4 in. (± 6 mm) at lower pressures. The input shadowgraph patterns are time-averaged by the NTSC camera and differ between fields of a frame. Definitive control is difficult when the plenum pressure P_7 is less than about 9 psia (62 kilopascal).

Plenum pressures greater than about 19 psia (131 kilopascal) correspond to a fully developed flow region where the input shadowgraph changed very slowly, if at all, with pressure. This region showed strong, stable, oblique shock waves originating from the nozzle exit The operator has reported sensing shadowgraph contrast changes as inlet pressure was increased in the fully developed flow region. But the feedforward nets are not especially sensitive to contrast or brightness changes as has been reported before.[2] Hence, the inability of the nets to estimate sensor values from shadowgraph patterns at the higher pressures is also understandable.

Part of the sequence of training images used to train the image-to-image sequencer has already been shown in Fig. 7. The image-to-image net learned this sequence very well at plenum pressures greater than 9 psia (131 kilopascal). The top row of Fig. 10 shows these training outputs again, and the bottom row of Fig. 10 shows the corresponding net generated images. Referring to Table I, the images correspond left-to-right to the transitions for training state numbers 6, 7, 8, and 9.

Figure 11 shows the performance of the image-to-image sequencer for some of the test images. The top row shows the actual shadowgraph images, and the bottom row shows the corresponding net generated images. Keep in mind that the shadowgraph patterns are quite sensitive to changes in the sensor values and the valve settings. There are no identical transitions in Tables I and II. The training images cover the space of possible images quite sparsely. And the dirt patterns on the windows differ between the training and test runs. Referring to Table II, the images correspond left-to-right to the transitions for test state numbers 5, 6, 7, and 8.

Figure 12 shows the performance of the sensor-to-sensor sequencer. The current and next sensor values in Tables I and II are complete input-output training and test records, respectively, for this sequencer. The training performance in the upper row is not perfect, but the test performance in the lower row is comparable, at least when compared with the performances of the image trained nets. The sensor-to-sensor net was definitely able to learn by example that the ratio of sensor values is an approximate invariant; the net functions even in the unstable flow region and to some extent in the fully developed flow region.

CONCLUDING REMARKS

Figures 8-12 show that a tunnel operator can train by example artificial neural networks to estimate the next operating state of a tunnel, but the process is by no means straightforward. The operator's intuition and personal anecdotes have little value in designing the process. First, one must identify exactly the sensor and flow visualization data that the operator uses in controlling operations. The operator of the supersonic wind tunnel clearly used very few sensors while operating the tunnel to generate the training sets for this paper. Nevertheless, as stated, the neural nets could not learn to estimate the few raw sensor readings. Products of sensor readings were required to train the nets successfully. This effect was recognized some time ago by Yoh-Han Pao and is the rational behind his proprietary functional link net.[12] The tunnel operator evidently was able to work with individual sensors without consciously assembling the readings into invariants. The standard feedforward nets were not able to learn from this example. The conclusion is that preprocessing with an assembling algorithm is required. An alternative is to have good constraining models which indicate the invariants or approximate invariants for training the nets.

Flow visualization images were also used successfully to estimate the next sensor values to be set. The sensor values again must be preassembled into the proper products and quotients for successful training. The procedures for handling the flow visualization images themselves require more development. There are interesting and useful questions to be answered. Are there simple preprocessing steps, akin to selecting sensor invariants for example, that would improve training? Can model generated images be used for training? One indication that model generated images might be very effective is that feedforward nets show low contrast sensitivity. The consequence is that model images, not differing much from line drawings, might be suitable for training artificial neural networks for tunnel control. The nets trained with these model images might not be affected by the variations in brightness, the variations in contrast, the addition of noise, or the blurring of edges found in real flow visualization images. The availability of good tunnel models would again help.

The overall project for testing neural net control of wind tunnel operations has resulted in some good suggestions for introducing, evaluating and managing automation of wind tunnel operations. The fast parallel processing workstation is to be used essentially as a slave module. In effect, the workstation maps the current state of the tunnel onto the next state of the tunnel and deposits the results in a file. The tunnel control system simply picks up and uses the results when ready. As mentioned, the SGI to Modicon interface is being developed with this protocol in mind. The protocol allows different approaches to automation to be tested in any facility while minimizing interruption of the facility's operations or disturbances of the facility's configurations.

The artificial neural networks can measure the effectiveness and consistency of utilization of the tunnel's resources. The tests performed in connection with this project isolated the few sensor values actually used to control the operation of the tunnel. Anecdotal claims of relationships between sensor values, flow visualization features, responses of controls, effectiveness of models and tunnel operations are suspicious when the neural nets are unable to learn these relationships.

REFERENCES

1) Decker, A., and Buggele, A., "Wind Tunnel Operations Using Archival Flow Visualization Records and Artificial Neural Networks," AIAA Paper 94-0390, Jan. 1994.

2) Buggele, A. E., and Decker, A. J., "Control of Wind Tunnel Operations Using Neural Net Interpretation of Flow Visualization Records," NASA TM 106683, Aug. 1994.

3) MODICON, vended by MODICON AEG, 1 High St., North Andover, MA 01845.

4) Brinich, P. F., "Boundary-Layer Measurements in 3.84- By 10-Inch Supersonic Channel," NACA Technical Note 2203, Oct 1950.

5) Crimson XS24 Workstation and VideoFramer, vended by Silicon Graphics, Inc., 2011 N. Shoreline Boulevard, Mountain View, CA 94043.

6) COHU Model 4815 Solid-State Monochrome CCD Camera, vended by COHU, Inc., 5755 Kearny Villa Road, San Diego, CA 92123.

7) NeuralWorks Professional II/PLUS, vended by NeuralWare, Inc., Penn Center West, Building IV, Pittsburgh, PA 15276.

8) Rumelhart, D. E., McClelland, J. L., and the PDP Research Group, *Parallel Distributed Processing--Explorations in the Microstructure of Cognition, Volume I: Foundations*, MIT Press, Cambridge, Mass., 1986.

9) Pao, Y-H., *Adaptive Pattern Recognition and Neural Networks*, Addison-Wesley, New York, 1989.

10) *Proceedings of the Second Workshop on Neural Networks: Academic/Industrial/NASA/Defense*, The Society for Computer Simulation, San Diego, 1991.

11) *Proceedings of the Third Workshop on Neural Networks: Academic/Industrial/NASA/Defense*, The Society for Computer Simulation, San Diego, 1992.

12) Reference 8, pp. 197-222.

Table I. Sensor Outputs (Training)

	Current			Next		
State	P_7 psi	$(MN_3)^2P_8$ psi	Ratio r	P_7 psi	$(MN_3)^2P_8$ psi	Ratio r
1	7.73	2.60	0.337	8.98	3.34	0.372
2	8.98	3.34	0.372	9.80	3.83	0.391
3	9.80	3.83	0.391	10.49	4.67	0.445
4	10.49	4.67	0.445	11.76	4.74	0.403
5	11.76	4.74	0.403	12.79	4.93	0.385
6	12.79	4.93	0.385	13.92	4.98	0.358
7	13.92	4.98	0.358	15.11	5.32	0.352
8	15.11	5.32	0.352	16.44	5.93	0.361
9	16.44	5.93	0.361	19.15	6.87	0.359
10	19.15	6.87	0.359	22.09	8.06	0.365
11	22.09	8.06	0.365	26.84	9.95	0.371

P_7 is the static inlet pressure.
P_8 is a static test section pressure.
MN_3 is the mach number downstream of the shadowgraph.
r is the ratio $(MN_3)^2P_8/P_7$.

Table II. Sensor Outputs (Test)

State	P_7 psi	Current $(MN_3)^2P_8$ psi	Ratio r	P_7 psi	Next $(MN_3)^2P_8$ psi	Ratio r
1	6.26	3.14	0.502	8.29	3.43	0.414
2	8.29	3.43	0.414	9.30	3.79	0.408
3	9.30	3.79	0.408	10.36	3.87	0.374
4	10.36	3.87	0.374	11.36	3.98	0.350
5	11.36	3.98	0.350	12.53	4.26	0.340
6	12.53	4.26	0.340	13.47	4.73	0.351
7	13.47	4.73	0.351	14.29	5.08	0.355
8	14.29	5.08	0.355	18.25	6.33	0.347
9	18.25	6.33	0.347	19.45	6.76	0.348
10	19.45	6.76	0.348	20.50	7.13	0.348
11	20.50	7.13	0.348	22.30	7.95	0.356
12	22.30	7.95	0.356	23.41	7.69	0.328

P_7 is the static inlet pressure.
P_8 is a static test section pressure.
MN_3 is the mach number downstream of the shadowgraph.
r is the ratio $(MN_3)^2P_8/P_7$.

FIGURE CAPTIONS

Fig. 1 1946 configuration of wind tunnel.[4]

Fig. 2 Modernized mach 2 configuration of wind tunnel.

Fig. 3 Schematic of 1994 configuration of tunnel services.

Fig. 4 Tunnel components and shadowgraph field.

Fig. 5 Flow of information handling and control.

Fig. 6 Feedforward neural network. Connections shown for one input and one output node.

Fig. 7 Part of a sequence of raw (top row) and processed (bottom row) shadowgraph patterns from the image controlled run. The flow is from right to left.

Fig. 8 Performance of image-to-sensor neural network in predicting current sensor outputs. Top row shows response to training images; bottom row shows response to test images. Lines represent measured outputs; dots represent net generated outputs. State numbers are from Tables I and II.

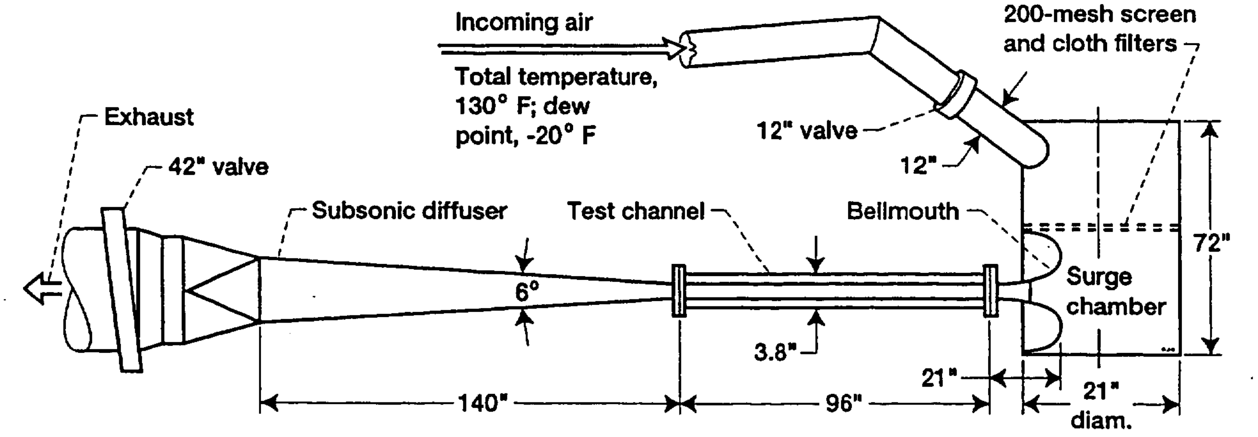
Fig. 9 Performance of image-to-sensor neural network as sequencer for predicting next sensor outputs. Top row shows response to training images; bottom row shows response to test images.

Fig. 10 Performance of image-to-image neural network as sequencer for predicting next shadowgraph pattern. Top row shows actual training shadowgraph patterns; bottom row shows corresponding net generated patterns.

Fig 11 Performance of image-to-image neural network as sequencer for predicting next shadowgraph pattern. Top row shows actual test shadowgraph patterns; bottom row shows corresponding net generated patterns.

Fig. 12 Performance of sensor-to-sensor neural network as sequencer. Top row shows performance for training records; bottom row shows performance for test records.

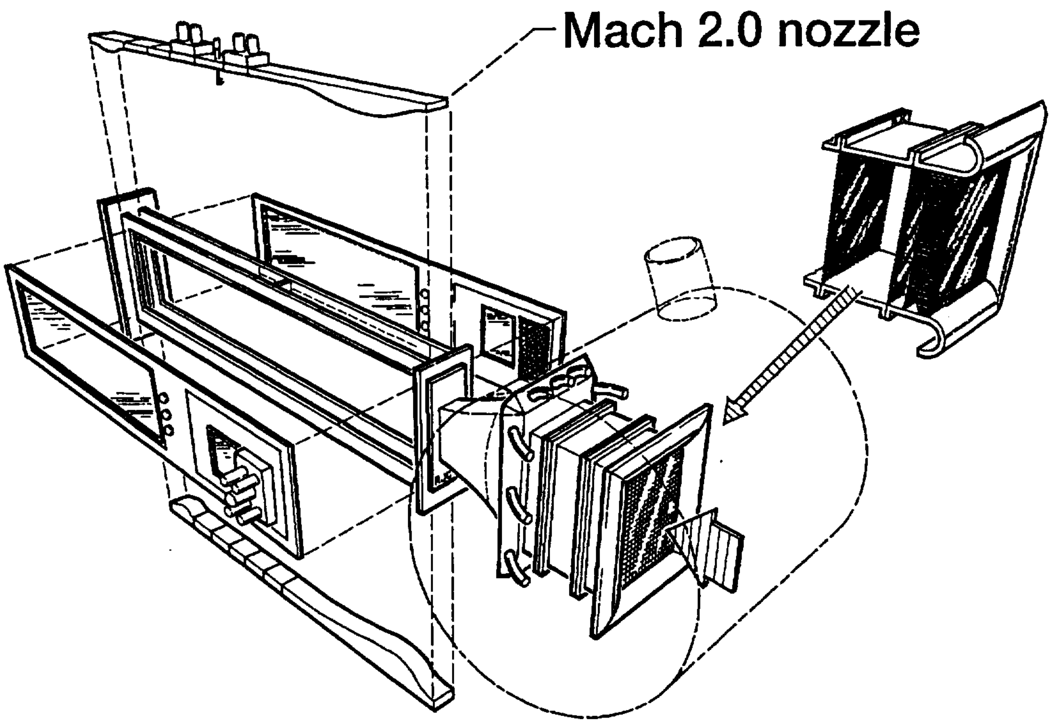
Arthur J. Decker
Fig. 1



CD-95-71012

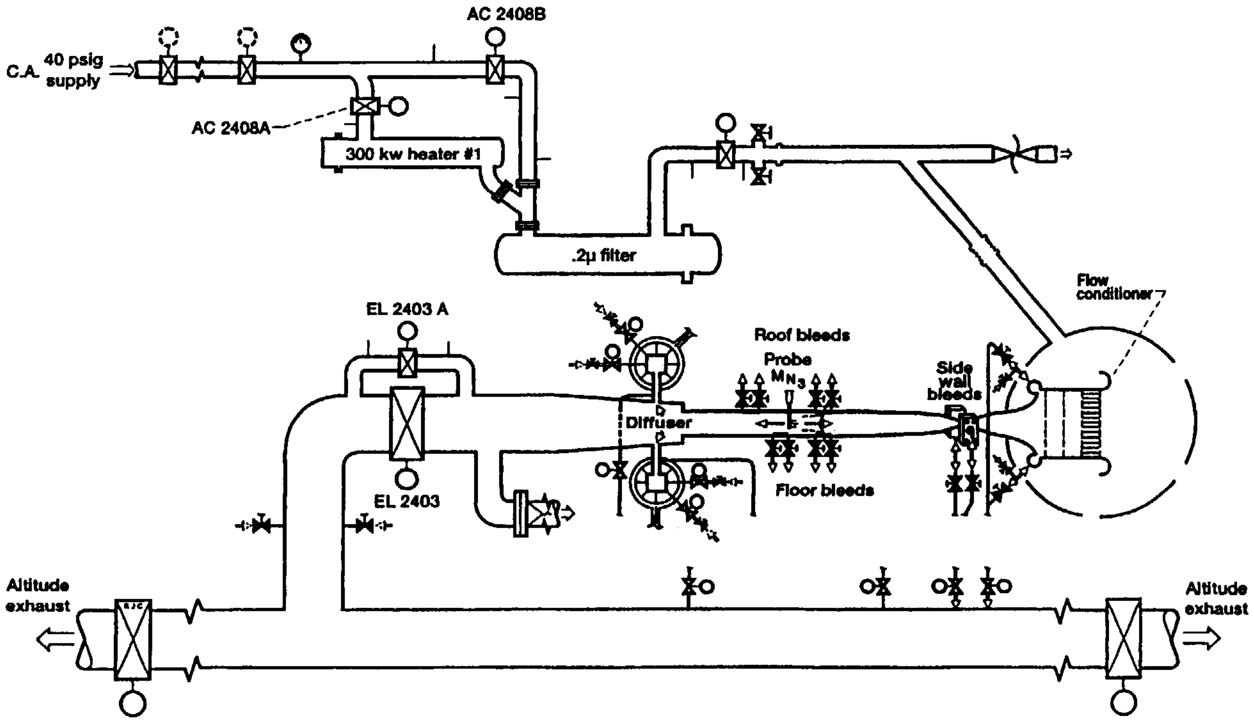
Arthur J. Decker

Fig. 2



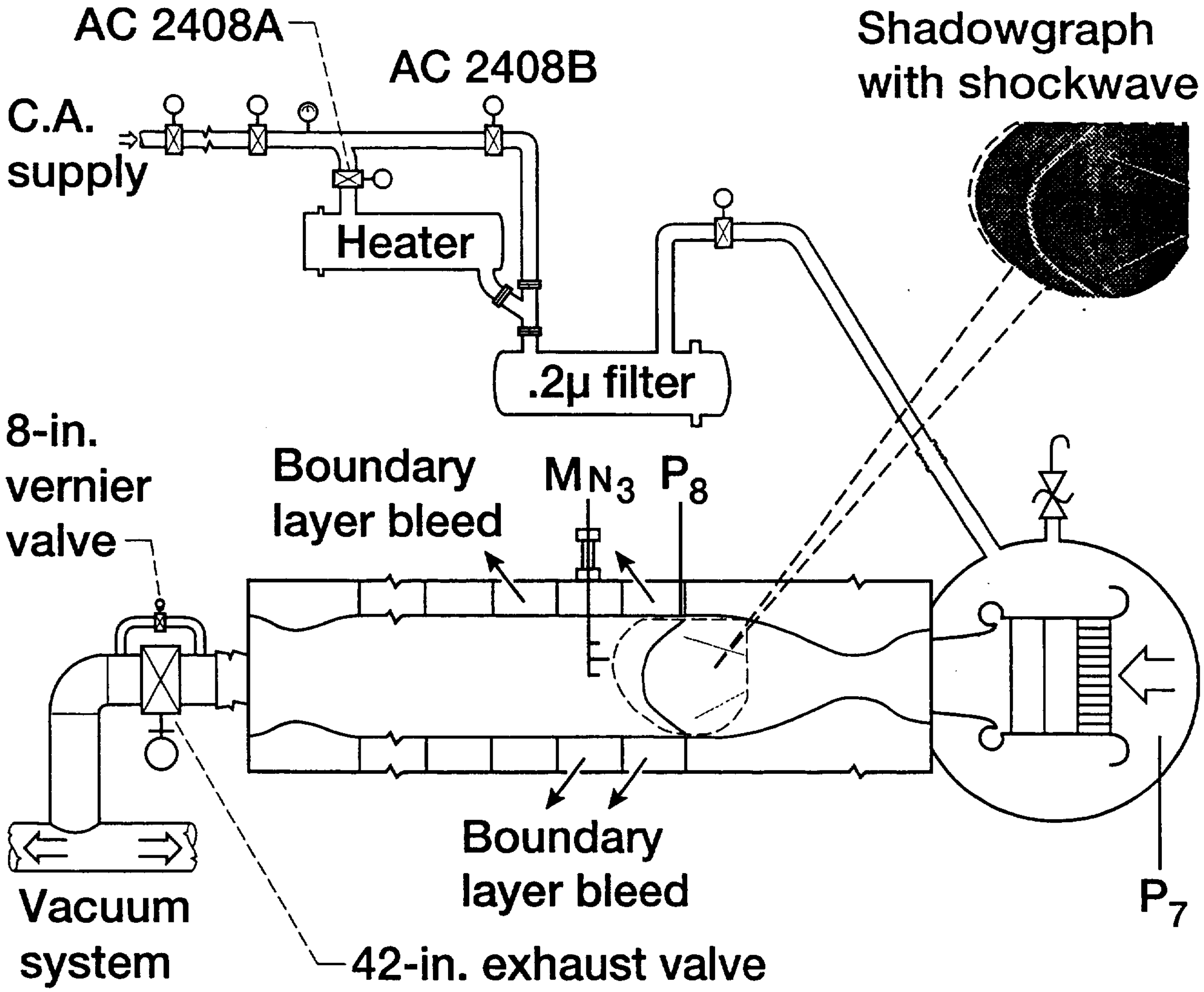
CD-95-71021

fig. 3



CD-95-71114 (Buggele, A.)

rthur J. Decker
Fig. 4



CD-95-71115

Fig. 5

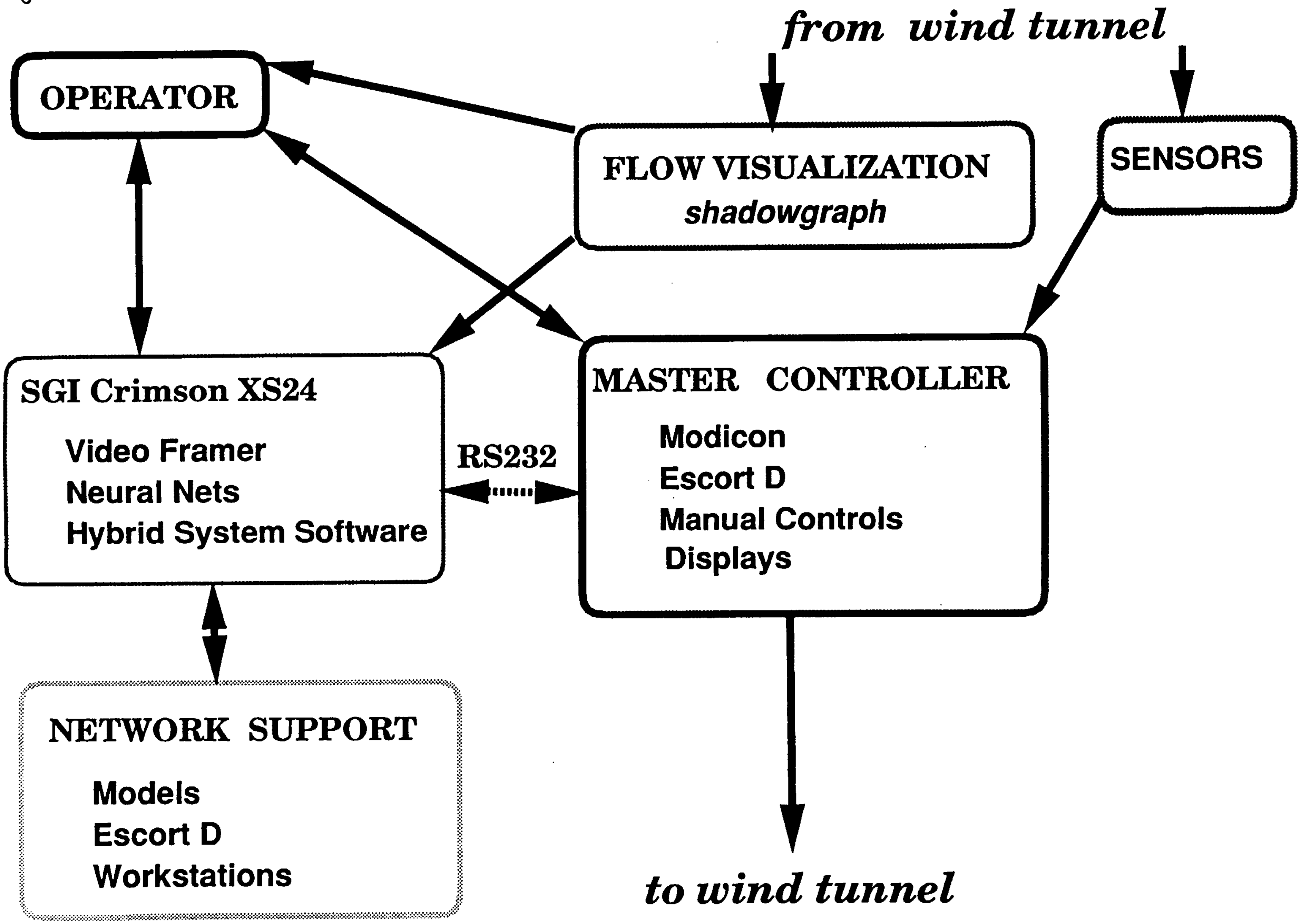

from wind tunnel
OPERATOR
FLOW VISUALIZATION
shadowgraph
SENSORS
SGI Crimson XS24
Video Framer
Neural Nets
Hybrid System Software
RS232
MASTER CONTROLLER
Modicon
Escort D
Manual Controls
Displays
NETWORK SUPPORT
Models
Escort D
Workstations
to wind tunnel

Fig. 6

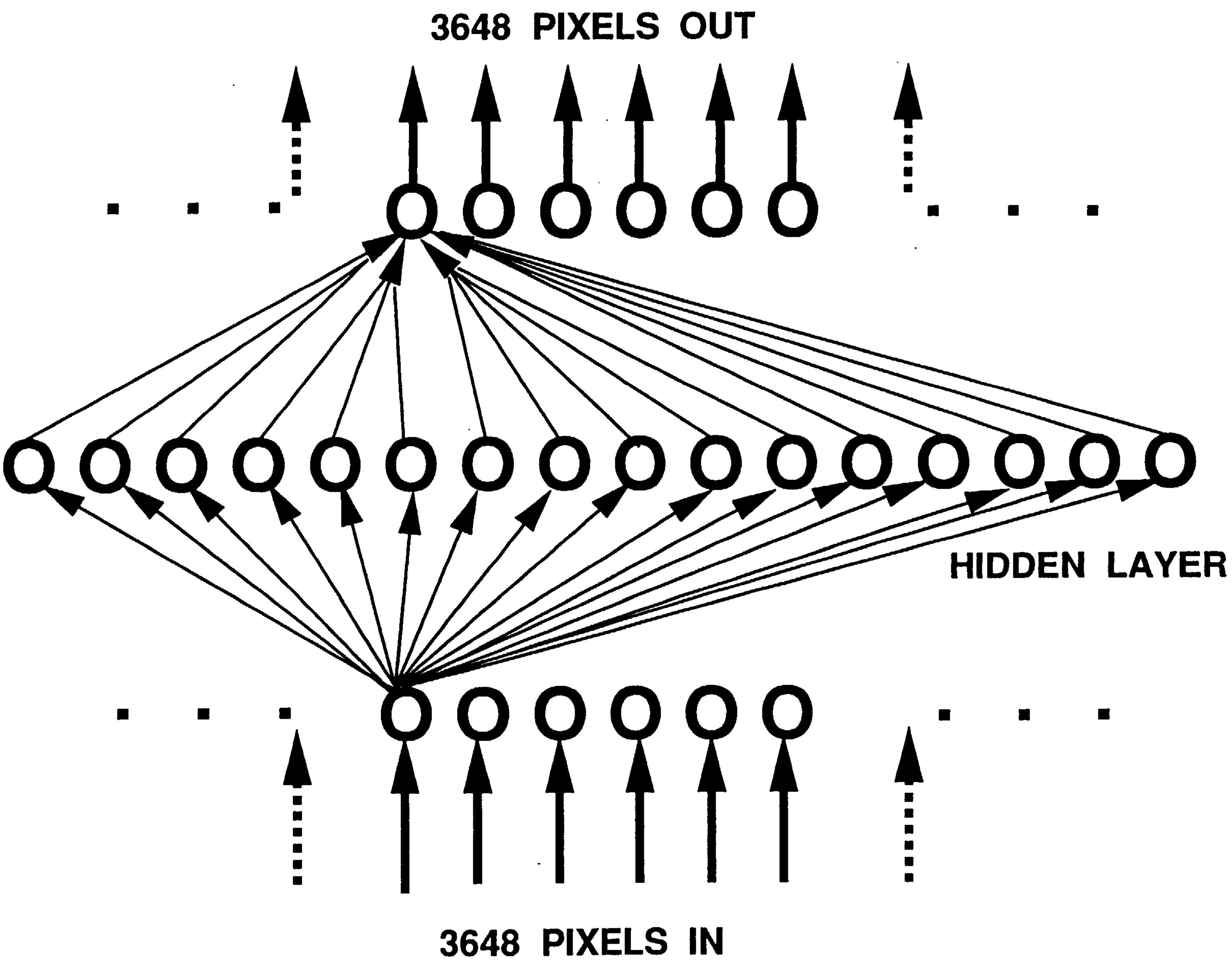
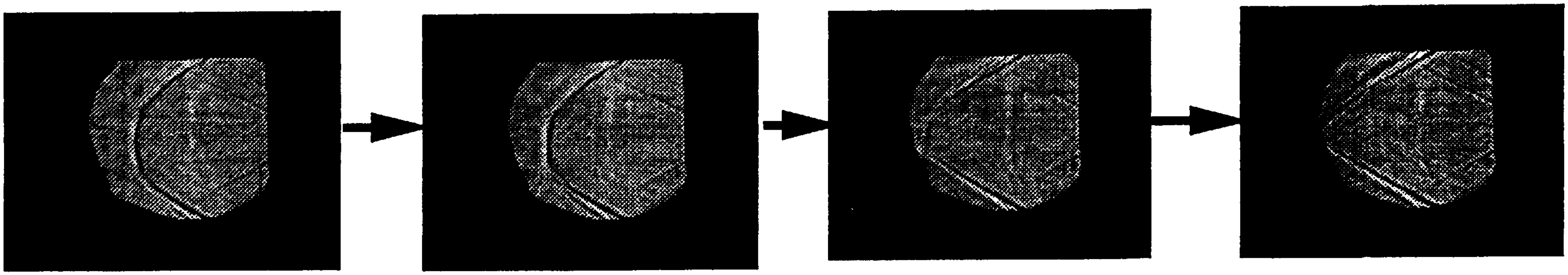
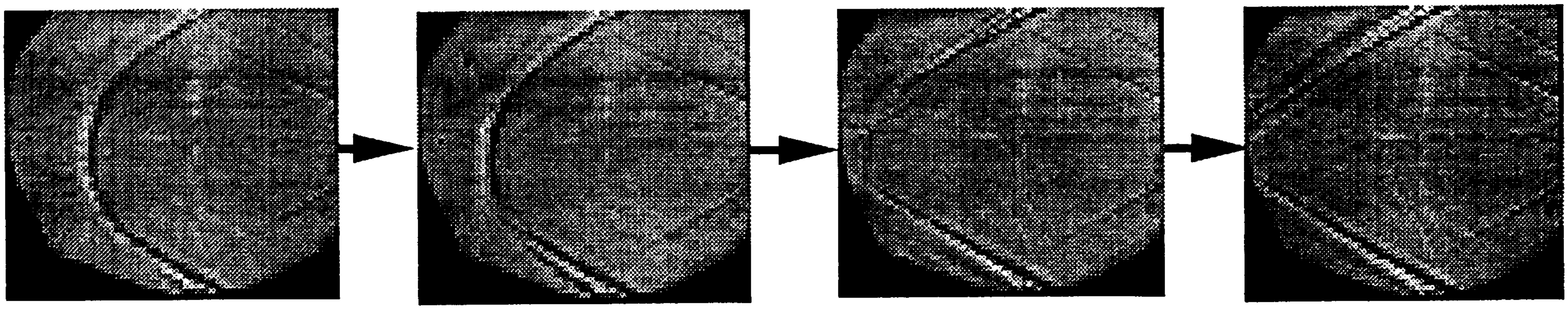


Fig. 7



PART OF SEQUENCE FROM 646 X 486 PIXEL CAMERA



SEQUENCE CROPPED AND REDUCED TO 64 X 57 PIXELS

Fig. 8

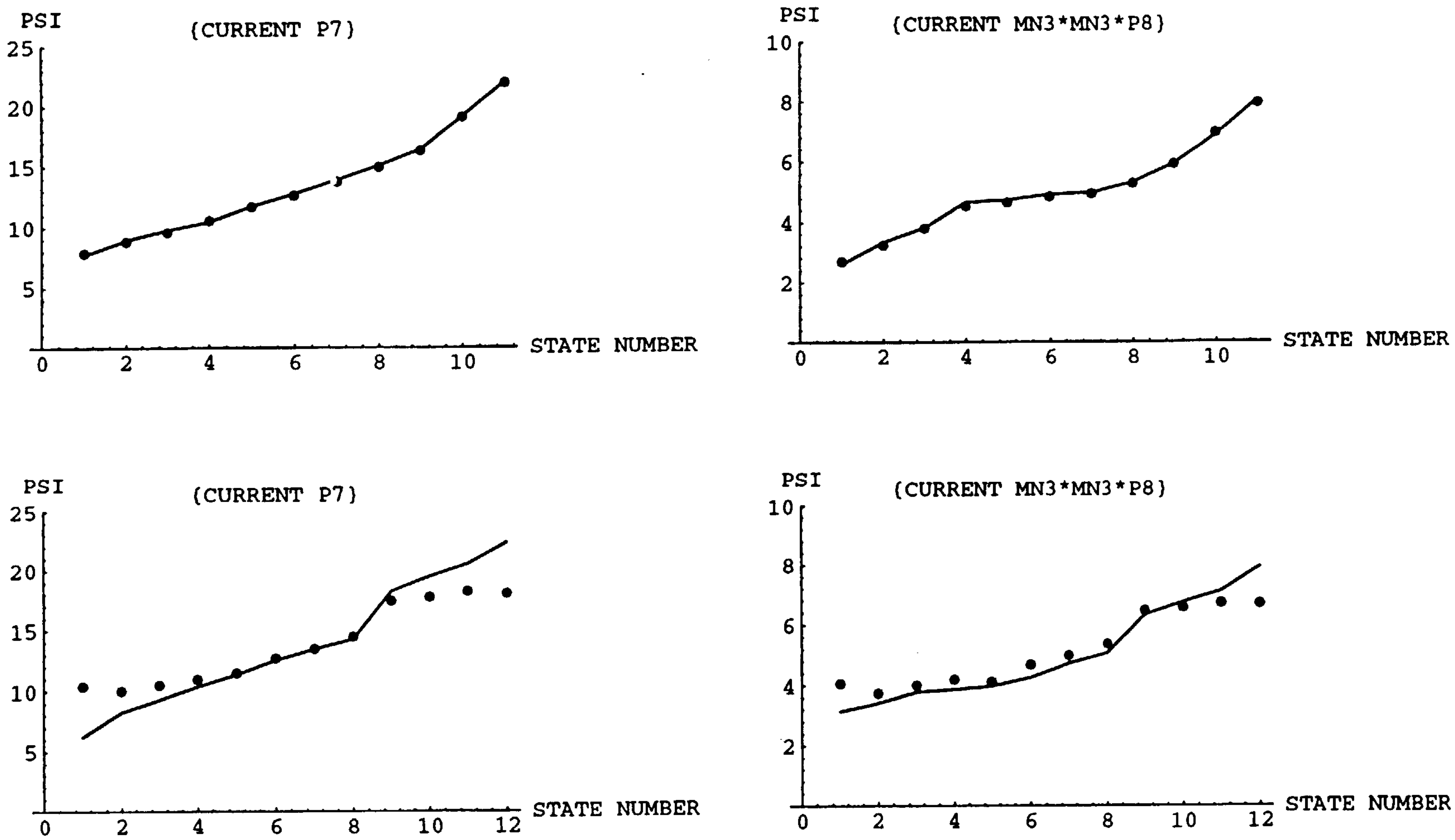

PSI
(CURRENT P7)
25
20
15
10
5
0
2
4
6
8
10
STATE NUMBER
PSI
(CURRENT MN3*MN3*P8)
10
8
6
4
2
0
2
4
6
8
10
STATE NUMBER
PSI
(CURRENT P7)
25
20
15
10
5
0
2
4
6
8
10
12
STATE NUMBER
PSI
(CURRENT MN3*MN3*P8)
10
8
6
4
2
0
2
4
6
8
10
12
STATE NUMBER

Fig. 9

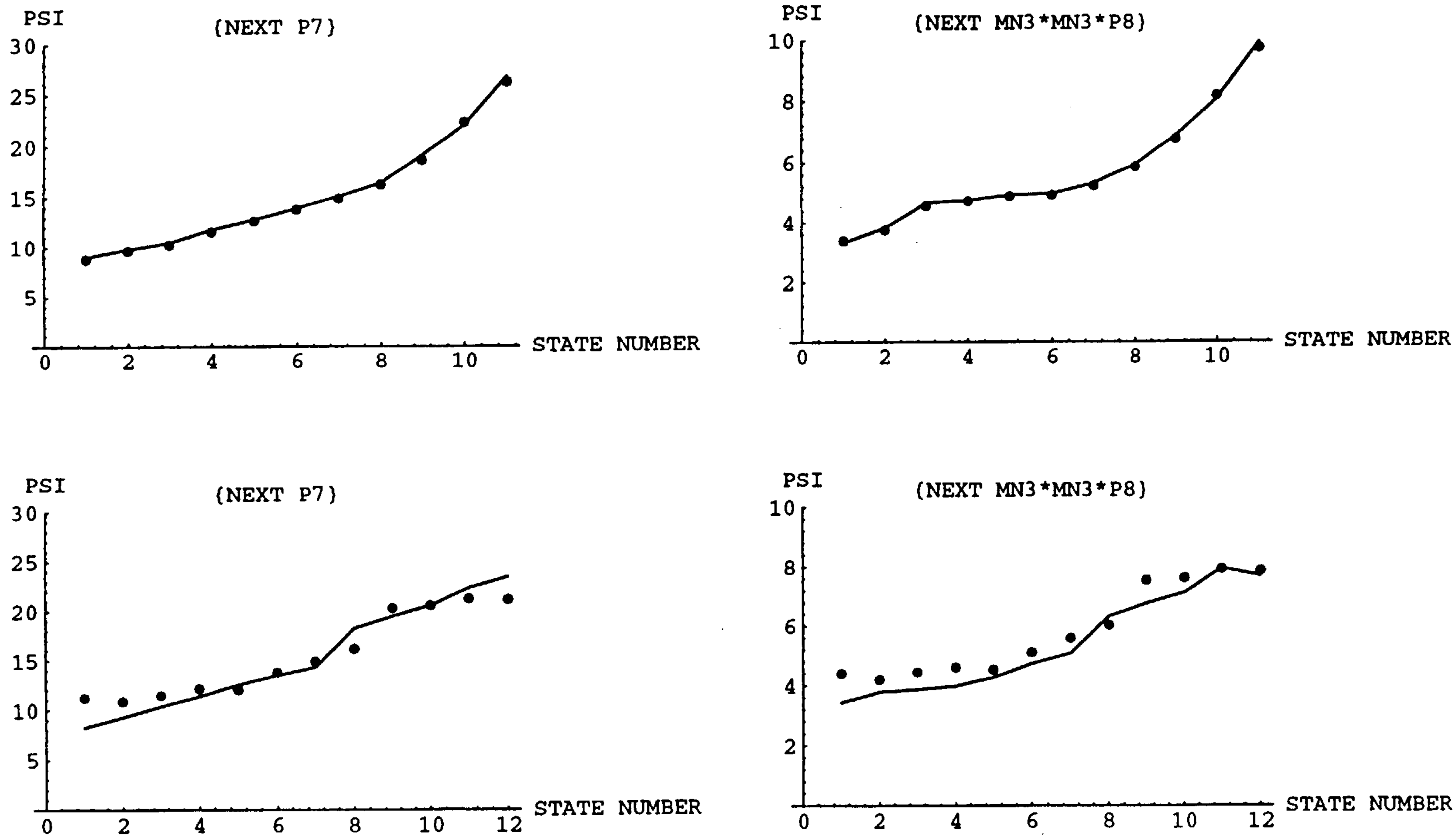

PSI
{NEXT P7}
30
25
20
15
10
5
0
2
4
6
8
10
STATE NUMBER
PSI
{NEXT MN3*MN3*P8}
10
8
6
4
2
0
2
4
6
8
10
STATE NUMBER
PSI
{NEXT P7}
30
25
20
15
10
5
0
2
4
6
8
10
12
STATE NUMBER
PSI
{NEXT MN3*MN3*P8}
10
8
6
4
2
0
2
4
6
8
10
12
STATE NUMBER

Fig. 10

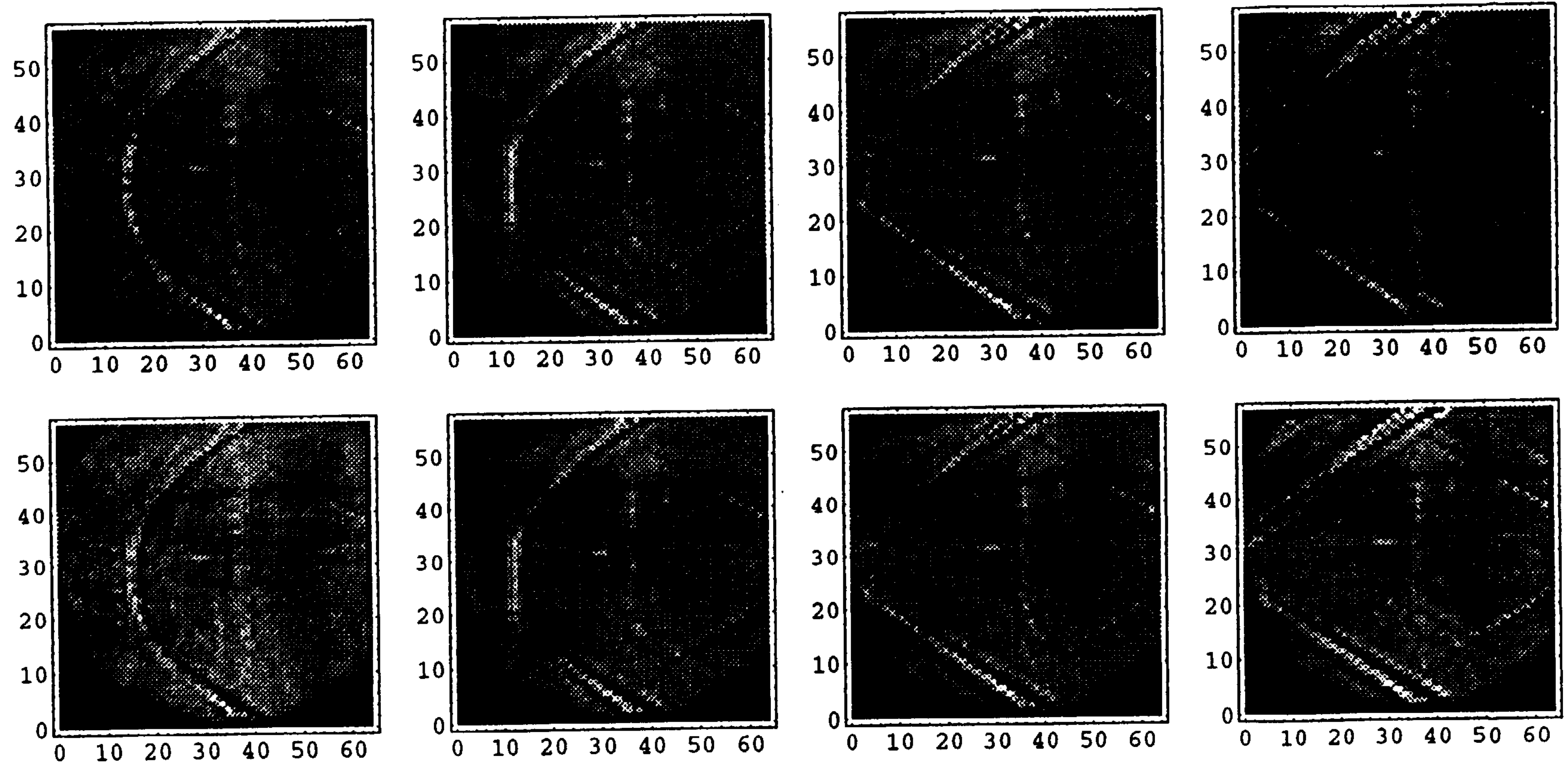

0
10
20
30
40
50
0 10 20 30 40 50 60
0
10
20
30
40
50
0 10 20 30 40 50 60
0
10
20
30
40
50
0 10 20 30 40 50 60
0
10
20
30
40
50
0 10 20 30 40 50 60
0
10
20
30
40
50
0 10 20 30 40 50 60
0
10
20
30
40
50
0 10 20 30 40 50 60
0
10
20
30
40
50
0 10 20 30 40 50 60
0
10
20
30
40
50
0 10 20 30 40 50 60

Fig. 11

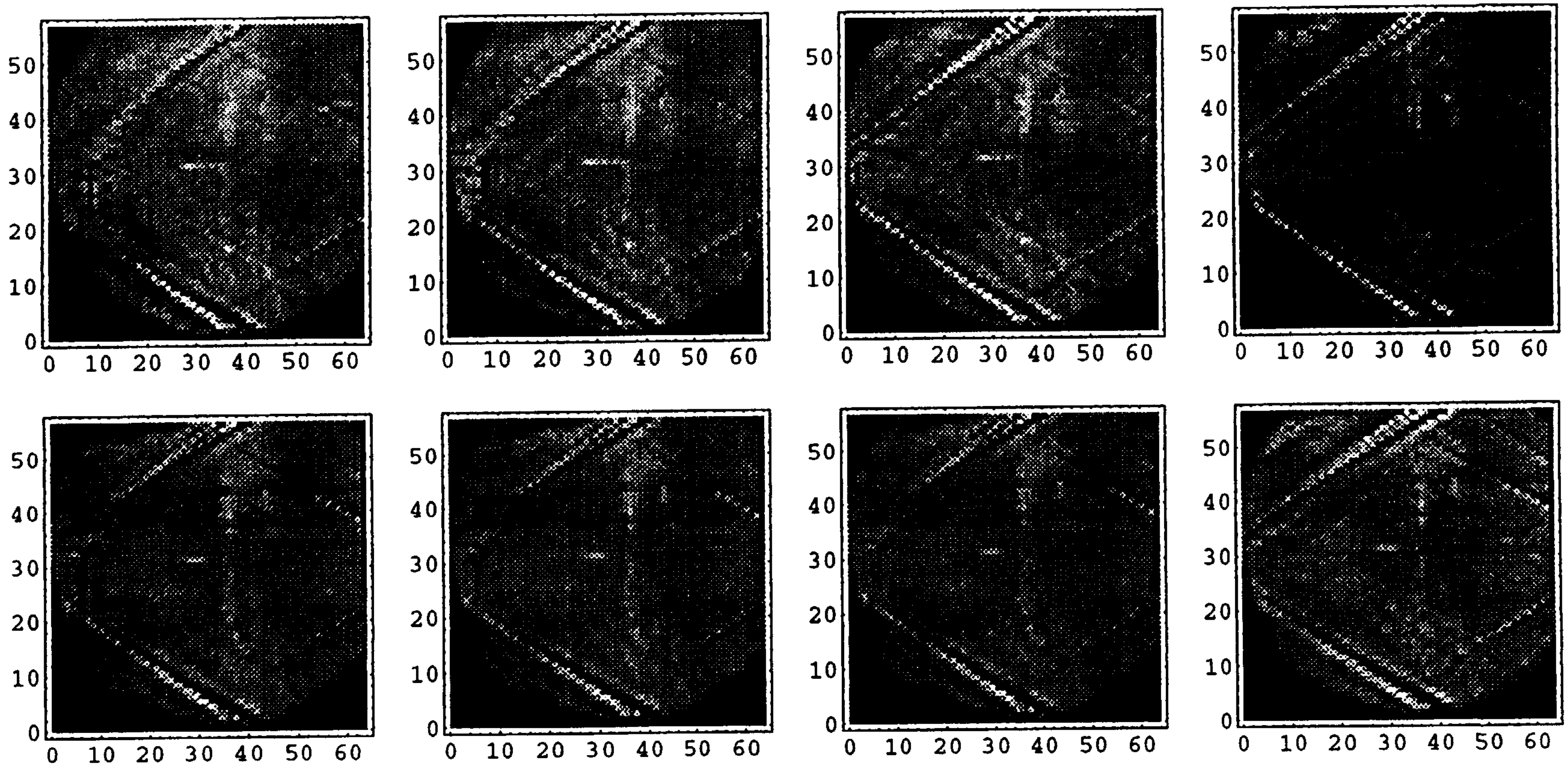
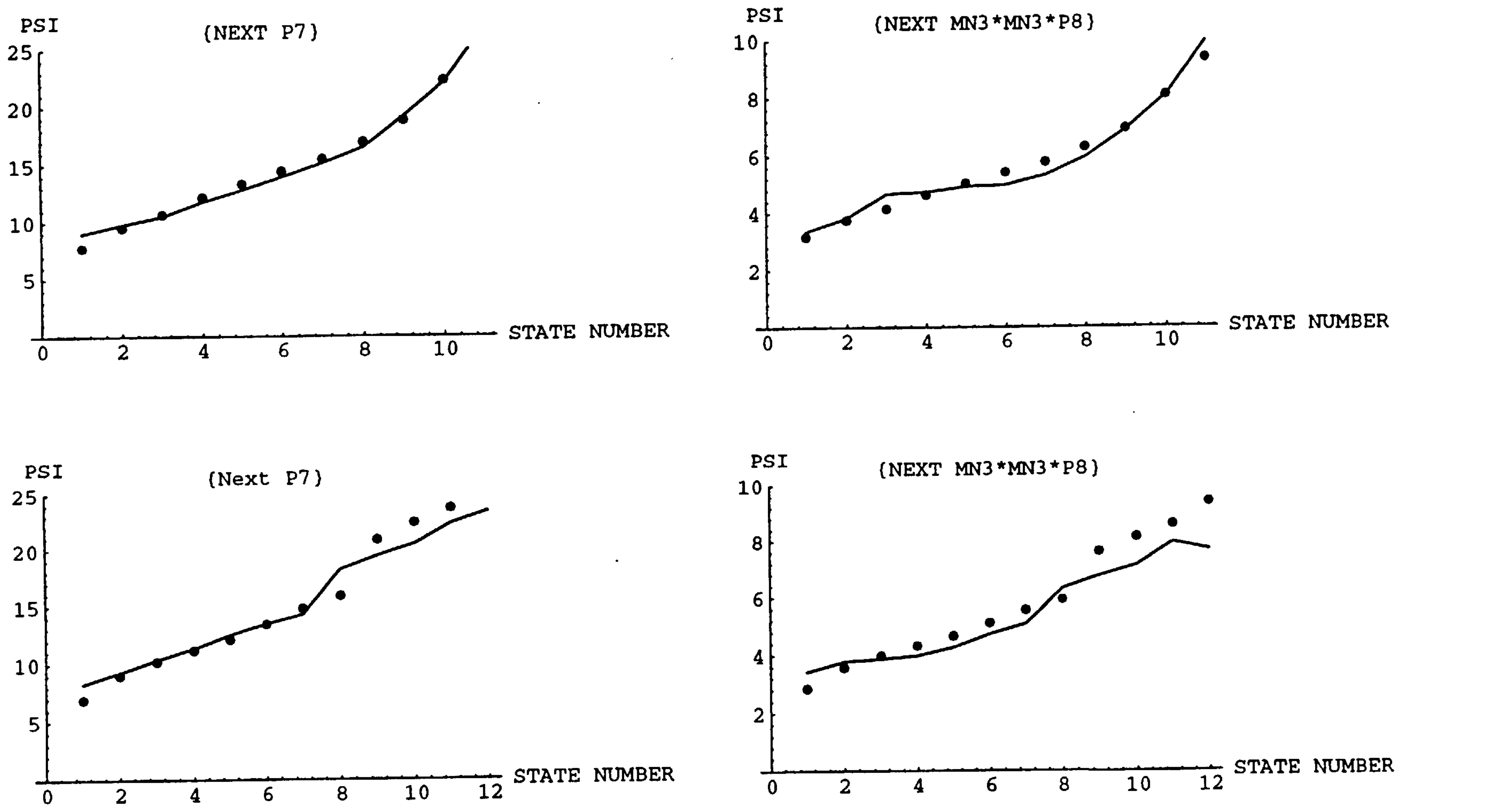


Fig. 12



(NEXT P7)

(NEXT MN3*MN3*P8)

(Next P7)

(NEXT MN3*MN3*P8)